

JP-A-11-8854

[0080]

Next, the second embodiment of a cut point detection unit 19 will be explained. In the second embodiment, it is determined that a cut point is present in the case where two change amounts obtained by two change amount calculation units (14, 17) are multiplied by weight and the sum of the two is equal to or greater than a threshold value.

10 [0081]

The second embodiment of the cut point detection unit 19 will be explained in detail with reference to Fig. 11. At step 400, the maximum value $\max \{DP_{h,k}\} (k=0,1,2,\dots,13)$ of a calculated change amount is determined from P picture row 13 of a certain group h. At step 401, it is determined whether a cut point is present in the group h. Then, the maximum value $\max \{DP_{h,k}\}$ of a change amount calculated from P picture row 13 of a certain group h and the change amount DI_{h+1} of I picture row 16 of the next group h+1 are multiplexed by weighs WP and WI, respectively ($WP=14/15$, $WI=1/15$), thereby obtaining the sum of the two. In the case where the sum is greater than a threshold T (Yes), it is determined at step 402 that a cut point is present in the group h. In the case where it is determined that

a cut point is present, the change amounts $WP * \max\{DP_{h,k}\}$ and $WI * DI_{h+1}$, which are multiplexed by the weights, are compared to each other at step 403 to specify a frame number. If the change amount $WP * \max\{DP_{h,k}\}$ is greater(Yes), it is determined at step 405 that a cut point is present at a frame number k' corresponding to $\max\{DP_{h,k}\}$. Further, when $WI * DI_{h+1}$ is greater, it is determined at step 406 that a cut point is present immediately before $WI * DI_{h+1}$.

10 [0082]

In the case where it is determined at step 401 that a cut point is not present in the group h (No), processes terminate at step 404.

[0083]

15 The weight and threshold to be used in the present invention can be predetermined in advance or can vary. How to vary the weight and threshold will be described later.

[0102]

20 (Fourth embodiment)

The present invention comprises a hard disk and the like for storing compressed and coded data and freely reading out the data, a buffer and the like required at the time of processing data, and a display and the like for displaying and outputting a cut point which

25

is finally detected. Further, using a computer or the like provided with a central processing unit for controlling the hard disk, buffer, display, etc., according to predetermined procedures, the present invention can perform the processes shown in Figs. 1,2,3,8,9,10 and 11 by providing the computer with the methods or algorithms described in the first, second and third embodiments, thereby causing the computer to control or carry out the processes. Please note that the process programs and the like can be stored in a floppy disk (FD), a magneto-optical disk (MO) or a similar storage medium which can be read out by the computer when the computer performs processes.

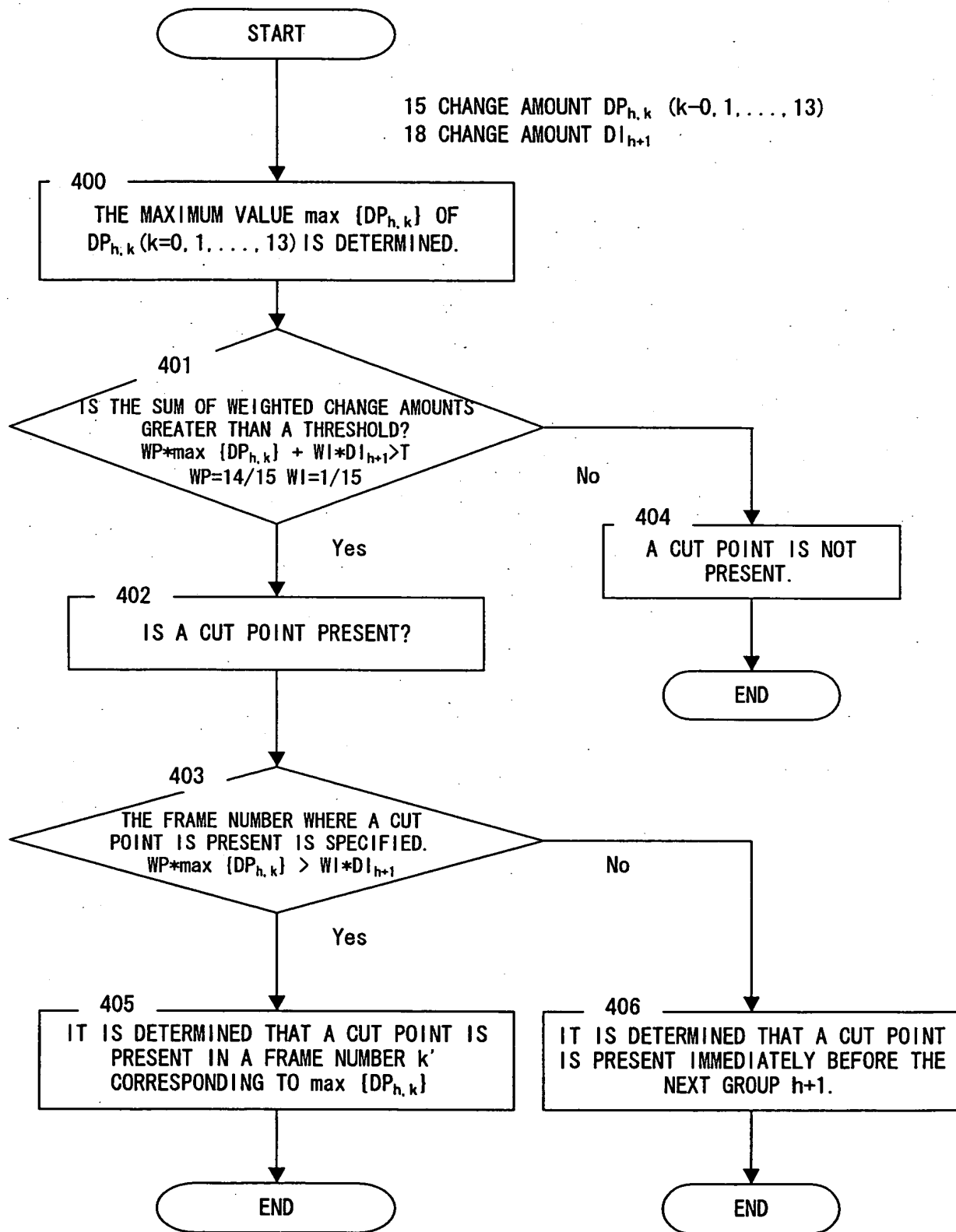


FIG. 11

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-8854

(43) 公開日 平成11年(1999) 1月12日

(51) Int.Cl.⁸

H 0 4 N 7/32

識別記号

F I

H 0 4 N 7/137

Z

審査請求 有 請求項の数10 O L (全 21 頁)

(21) 出願番号 特願平9-108601

(22) 出願日 平成9年(1997) 4月25日

(31) 優先権主張番号 特願平8-108858

(32) 優先日 平8(1996) 4月30日

(33) 優先権主張国 日本 (J P)

(31) 優先権主張番号 特願平9-105483

(32) 優先日 平9(1997) 4月23日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72) 発明者 新倉 康巨

東京都新宿区西新宿三丁目19番2号 日本

電信電話株式会社内

(72) 発明者 谷口 行信

東京都新宿区西新宿三丁目19番2号 日本

電信電話株式会社内

(72) 発明者 阿久津 明人

東京都新宿区西新宿三丁目19番2号 日本

電信電話株式会社内

(74) 代理人 弁理士 秋田 収喜

最終頁に続く

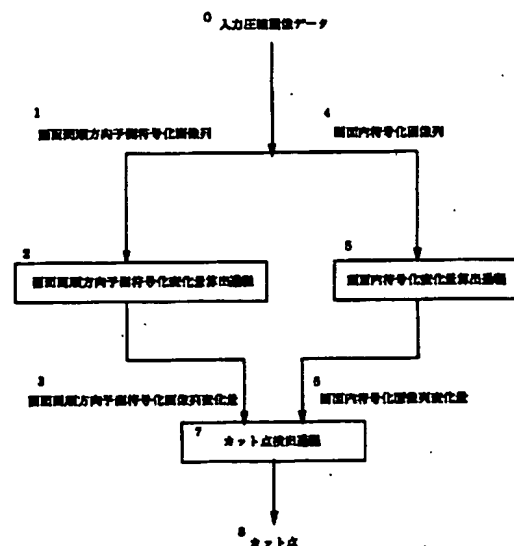
(54) 【発明の名称】 カット点検出方法及び装置

(57) 【要約】

【課題】 圧縮符号化映像からカット点を高速、高精度に検出する。

【解決手段】 画面間予測符号化方式と画面内符号化方式を共に含む圧縮符号化方式によって圧縮された画像データ列からカット点を検出するカット点検出方法において、画面間順方向予測符号化方式で圧縮された画像列 P_t に含まれる符号化情報を基に、画像列 P_t の変化量 DP_t を算出する画面間順方向予測符号化変化量算出過程2と、画面内符号化方式で圧縮された画像列 I_t に含まれる符号化情報を基に、画像列 I_t の変化量 DI_t を算出する画面内符号化変化量算出過程5と、両過程で得られる二つの変化量 DP_t と DI_t の両方を評価しカット点を検出するカット点検出過程7を具備するものである。

図 1



【特許請求の範囲】

【請求項1】 画面間予測符号化方式と画面内符号化方式を共に含む圧縮符号化方式によって圧縮された画像データ列からカット点を検出するカット点検出方法において、画面間順方向予測符号化方式で圧縮された画像列 P_t に含まれる符号化情報を基に、画像列 P_t の変化量 DP_t を算出する画面間順方向予測符号化変化量算出過程と、画面内符号化方式で圧縮された画像列 I_t に含まれる符号化情報を基に、画像列 I_t の変化量 DI_t を算出する画面内符号化変化量算出過程と、両過程で得られる二つの変化量 DP_t と DI_t の両方を評価しカット点を検出するカット点検出過程を具備することを特徴とするカット点検出方法。

【請求項2】 前記画面間順方向予測符号化変化量算出過程において、画面間予測が外れた領域が全面に占める割合を変化量とすることを特徴とする請求項1に記載のカット点検出方法。

【請求項3】 前記画面間順方向予測符号化変化量算出過程、画面内符号化変化量算出過程のいずれか一方又は両方で、画像列の隣り合う画像間で変化の生じた領域を変化領域として求める過程と、該変化領域の時間系列に対してあらかじめ与えられた空間的演算を施した後、それらを時間的に積分する過程を含むことを特徴とする請求項1に記載のカット点検出方法。

【請求項4】 前記変化量 DP_t と変化量 DI_t の両方を評価しカット点を検出するカット点検出過程において、画像列 P_t に含まれる符号化情報を基に得られた画像列 P_t の変化量 DP_t に対して、画像列 I_t から得られた情報を用いて、変化量 DP_t を補正する処理過程を含むことを特徴とする請求項1乃至3のうちいずれか1項に記載のカット点検出方法。

【請求項5】 前記変化量 DP_t と変化量 DI_t の両方を評価しカット点を検出するカット点検出過程において、画像列 I_t に含まれる符号化情報を基に得られた画像列 I_t の変化量 DI_t に対して、画像列 P_t から得られた情報を用いて、変化量 DI_t を補正する処理過程を含むことを特徴とする請求項1乃至3のうちいずれか1項に記載のカット点検出方法。

【請求項6】 前記カット点検出過程において、画面間順方向予測符号化変化量算出過程から得られる変化量 DP_t が、第一の閾値 TP_t を上回るか、画面内符号化変化量算出過程から得られる変化量 DI_t が、第二の閾値 TI_t を上回るか、どちらか一方が成立した場合に時刻 t にカット点が存在していると判定することを特徴とする請求項1, 4, 5のいずれか1項に記載のカット点検出方法。

【請求項7】 前記カット点検出過程において、画像列 P_t と画像列 I_t の出現頻度を測定する過程を含み、該出現頻度に応じて、二つの変化量 DP_t と DI_t を評価する手続きを切り替えることを特徴とする請求項1, 4, 5

のいずれか1項に記載のカット点検出方法。

【請求項8】 カット点検出過程において、画像列 P_t と画像列 I_t の出現頻度を測定する過程を含み、該出現頻度に応じて、二つの変化量 DP_t と DI_t に対する二つの閾値 TP_t , TI_t を制御することを特徴とする請求項1, 4, 5のいずれか1項に記載のカット点検出方法。

【請求項9】 コンピュータに、請求項1乃至8のいずれか1項に記載のカット点検出方法を機能させるための処理プログラムを記録したコンピュータが読み取り可能な記録媒体。

【請求項10】 画面間予測符号化方式と画面内符号化方式を共に含む圧縮符号化方式によって圧縮された画像データ列からカット点を検出するカット点検出装置において、画像データ列中の圧縮符号化方式の種類を判定し、符号化データを分類するピクチャ判定手段と、画面間順方向予測符号化方式で圧縮された画像列 P_t に含まれる符号化情報を基に、画像列 P_t の変化量 DP_t を算出するPピクチャ変化量算出手段と、画面内符号化方式で圧縮された画像列 I_t に含まれる符号化情報を基に、画像列 I_t の変化量 DI_t を算出するIピクチャ変化量算出手段と、両変化量算出手段で得られる二つの変化量 DP_t と DI_t を評価しカット点を検出するカット点検出手段とを具備することを特徴とするカット点検出装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、圧縮符号化映像データからカット点を検出するカット点検出方法及び装置に関し、特に、画面間予測符号化方式と画面内符号化方式を共に含む圧縮符号化方式によって圧縮された画像データ列からカット点を検出するカット点検出方法及び装置に適用して有効な技術に関するものである。

【0002】

【従来の技術】 映像データはデータ量が膨大であり、その内容を知るためには映像を時間順に全て見ていくしかなかった。映像がある基準に基づいて分割されていれば、映像を飛ばし見したり、内容を大雑把に把握するために役立つ。理想的には、ストーリーを考慮して分割する必要があるが、その作業は現状では人手でしかできず、作業量が膨大になるという問題点があり、なんらかの自動化が求められていた。そこで、連続して撮影された一つの場面であるショットという単位に分割する技術が必要である。

【0003】 映像を時間順に見ていったとき、ショットの切り替わった時点のカット点と呼ぶ。カット点前後では映像の内容が急激に変化するので、「時間順に隣り合う画像間の差を計算し、差が大きいところをカット点とみなす」という方法で、カット点を検出することができる。例えば、連続する画像フレームの位置(x, y)の輝度、色等の物理量が変化した画素数から連続するフレームの変化を算出し、カット点を検出する「動画のカッ

ト自動分割方法」(特開平5-37853号公報)等があげられる。

【0004】しかし、画像の内容が急激に変化する一般的なカット点に加えて、ワイプやディゾルブといった緩やかに変化するカット点を安定に検出し、カメラワークや動き被写体などによる画像変化や、フラッシュや高速で画面内を横切る物体等による画像の一時的な変化をカット点として誤検出することを防ぐことが課題であった。こうした課題を解決するカット点検出技術は既にいくつか存在する。しかしながら、これらのカット点検出技術は、アナログ映像信号をデジタル化した非圧縮映像データを対象としたものであった。

【0005】一方で、映像情報の利用が盛んになるにつれ、伝送や、蓄積メディアにおける負荷を軽減するため、映像の圧縮符号化技術が目覚ましい進歩を遂げており、H. 261, MPEG等の圧縮技術が標準化されている。こうした圧縮符号化方式では、画面内での冗長性を削減する画面内符号化方式と、画面間での冗長性を削減する画面間予測符号化方式の二つの方式が代表的である。

【0006】画面内符号化方式は、図16(a)に示すように、対象画像30を例えば正方形の複数のブロック31に分割し、それぞれのブロック31をDCT変換(離散コサイン変換)し、量子化することにより符号化する。この場合の符号化データは、ブロック31をDCT変換したDCT係数である。MPEG方式では画面内符号化データとして、ブロック31を輝度と色差成分に分解し、それぞれをDCT変換したDCT係数を保持している。

【0007】一方、画面間予測符号化方式では、図16(b)に示すように、対象画像30を正方形の複数のブロック31に分割し、それぞれのブロック31を過去の(異なる時間の)画像32の中で最も類似する領域33で代用し、符号化する方式である。隣接する画像の間では、画像の内容の変化は一般的に少ないので、ブロック31を類似領域33によって置き換えることで画面間の冗長性を削減することができる。このような方法を画面間予測符号化と呼び、注目ブロック31Aと類似領域33のずれを動きベクトル34と呼ぶ。

【0008】画面間予測符号化方式における符号化データは、動きベクトル34と、ブロック31と類似領域33の間の差分のDCT係数から構成されている。ブロック31に類似する領域が存在しない場合には、予測が外れたとし、ブロック31を画面内符号化方式と同様にDCT変換し、量子化して符号化する。このようなブロックをブロック内符号化ブロックと呼ぶ。

【0009】画面間予測符号化方式では、画面間順方向予測符号化方式と画面間双方向予測符号化方式の2つの方式が代表的である。画面間順方向予測符号化方式は、図16(b)に示すように、注目ブロック31Aに類似

した領域を過去の画像から探索する。画面間双方向予測符号化方式は、図16(c)に示すように、未来の画像37と過去の画像35の両方の画像から類似領域33を探索する。

【0010】従って、画面間順方向予測符号化方式のデータに含まれている動きベクトル34は、過去の画像の類似領域33と注目ブロック31Aとの位置のずれを表わし、画面間双方向予測符号化方式のデータに含まれている動きベクトル34は、ブロック31と過去の画像上での位置のずれか、未来の画像上での位置のずれかのどちらかのいずれか一方もしくは両方を表わしている。画面間順方向予測符号化方式によって符号化された対象画像30は、過去の画像32を復号化しなければ復号化できず、画面間双方向予測符号化方式によって符号化された対象画像30は、過去と未来の両方の画像35、37を先に復号化しなければ復号化できない。それに対して、画面内符号化方式で圧縮された画像はそれだけで復号化可能である。

【0011】MPEG方式は、画面内符号化方式と画面間順方向予測符号化方式、画面間双方向予測符号化方式を組み合わせた方式であり、今後の主流になるといわれている。MPEG方式では、画面内符号化方式、画面間順方向予測符号化方式、画面間双方向予測符号化方式によって圧縮された画像を、それぞれIピクチャ、Pピクチャ、Bピクチャと呼ぶ。MPEG方式による映像データでは、例えば、IBBPBBPBBPBBIBBPBBPBBPBBのように、各ピクチャが交互に出現する。ただし、その出現頻度は定まっておらず、同一データの中で様々に変動することが許される。

【0012】MPEG方式を始めとする圧縮符号化映像からカット点を検出するために、データを一度復号化して非圧縮デジタル映像を復元し、従来技術を用いてカット点を検出することは可能である。しかし、復号化処理は計算時間のかかる処理であるという問題点があった。

【0013】以下、復号化処理を行わずに、MPEG映像からカット点を検出する従来技術について述べる。

【0014】Boon-Lock Yeo and Bede Liu, IEEE Proceeding of the International Conference on Multimedia Computing and Systems, pp. 81-88に開示される「A Unified Approach to Temporal Segmentation of Motion JPEG and MPEG Compressed Video」は、IピクチャのDCT係数の直流成分と、Pピクチャ、Bピクチャの動きベクトルを用いて、各ピクチャの縮小画像を再構成し、その縮小画像を順次比較し、変化が大きいところをカット点として検出している。

【0015】しかし、縮小画像を再構成する部分的な復号化処理に時間がかかるという問題点があった。

【0016】Farshid Arman, Arding Hsu, and Ming-Ye Chiu, ACM Multimedia '93, pp.267-272)に開示され

10

20

30

40

50

る「Image Processing on Compressed Data for Large Video Database」、特開平 7-236153 号公報に開示されている「動画像のカット点検出およびカット画面グループ検出装置」は、I ピクチャの DCT 係数の比較によってカット点を検出している。

【0017】MPEG 方式では、一般的に I ピクチャの出現頻度は P や B ピクチャのそれに比べて低い。MPEG 方式の典型的な映像データ列では、I ピクチャは 1 秒間に 2 フレーム程度である。出現頻度が低いと、カメラの動きや被写体の動きをカット点として誤って検出してしまうことが多くなる。なぜなら、カメラの動きや被写体の動きがあると、次の I ピクチャが現われる前に大きく画像の内容が変化するためである。

【0018】従って、I ピクチャのみを用いたカット点検出技術は、I ピクチャの出現頻度が低くなればなるほど、誤検出率が高くなるという問題点があった。

【0019】特開平 4-207876 号公報に開示される「動画像管理装置」は、P ピクチャ中のブロック内符号化ブロックの個数を利用してカット点を検出する技術である。カット点があると画面間予測が外れるので、ブロック内符号化ブロックの個数が急増するという性質を利用している。

【0020】しかし、この方法では、P ピクチャ直前のカット点は検出可能だが、I ピクチャの直前に存在するカット点は検出できないという問題点がある。なぜなら、P ピクチャ生成時の予測において、注目画像に対する類似領域の探索対象画像は、注目画像直前の P ピクチャか I ピクチャであり、注目画像の直前の I ピクチャよりさらに過去の画像から類似領域を探索することはない。すなわち、I ピクチャ直前にカット点があることで、予測が外れ、ブロック内符号化ブロックの個数が増大した P ピクチャは存在しないからである。

【0021】Hong Jian Zhang, Hien Yong Low, Yihong Gong and Stephen W. Smoliar, SPIE, Vol. 2182, Image and Video Processing II, PP. 142-149, 1994 に開示される「Video Parsing Using Compressed Data」、特開平 7-284017 号公報に開示される「カット検出装置」は、B ピクチャのブロックが記録している動きベクトルが、過去と未来のどちらの画像を参照しているのかを調べることによってカット点を検出する。

【0022】しかし、B ピクチャの出現頻度は圧縮データ毎に多種多様であり、B ピクチャが全く出現しない圧縮映像データも存在する。このような圧縮映像データに対しては、カット点を全く検出することができないという問題点があった。

【0023】また、一般に、MPEG 符号化データにおいては、I ピクチャと P ピクチャのそれぞれの出現確率は、一般的には P ピクチャの方が I ピクチャよりも高い。従って、I ピクチャからの情報だけを基にした I ピクチャ変化量の特徴量は、大局的な変化を示し、P ピク

チャからの情報だけを基にした P ピクチャ変化量の特徴量は、細部に渡る変化を示すことになる。さらに、P ピクチャ特徴量では動き補償による情報を含んでおり、カメラワークや動き被写体の影響を受けづらいという特徴を有する。一方で、瞬間的な変化の影響を受けやすく、フラッシュや高速に横切る物体等による瞬時に発生するノイズをカット点として誤検出しやすいという問題があった。

【0024】

10 【発明が解決しようとする課題】本発明者は、前記従来技術を検討した結果、以下の問題点を見いだした。

【0025】(a) 圧縮データを復号化してカット点を検出すると、復号化に時間がかかってしまうこと、

(b) カメラや被写体の動きを誤ってカット点として検出してしまうこと、(c) フラッシュなどのノイズを誤ってカット点として検出してしまうこと、(d) ワイプやディゾルブ等の緩やかなカット点を検出しづらいこと、(e) P ピクチャ変化量からのカット点検出方法は、P ピクチャと I ピクチャの間に存在するカット点を検出することはできないこと、(f) P ピクチャ変化量からのカット点検出方法はフラッシュや高速に横切る物体等の瞬時に発生するノイズをカット点として誤検出しやすいこと、(g) 圧縮符号化データ中の各ピクチャの出現頻度によってカット点検出での精度が変化すること等である。

【0026】本発明の目的は、圧縮符号化映像からカット点を高速、高精度に検出することが可能な技術を提供することにある。

30 【0027】本発明の前記ならびにその他の目的と新規な特徴は、本明細書の記述及び添付図面によって明らかにする。

【0028】

【課題を解決するための手段】本願において開示される発明のうち代表的なものの概要を簡単に説明すれば、以下のとおりである。

40 【0029】(1) 画面間予測符号化方式と画面内符号化方式を共に含む圧縮符号化方式によって圧縮された画像データ列からカット点を検出するカット点検出方法において、画面間順方向予測符号化方式で圧縮された画像列 P_t に含まれる符号化情報を基に、画像列 P_t の変化量 DP_t を算出する画面間順方向予測符号化変化量算出過程と、画面内符号化方式で圧縮された画像列 I_t に含まれる符号化情報を基に、画像列 I_t の変化量 DI_t を算出する画面内符号化変化量算出過程と、両過程で得られる二つの変化量の両方を評価しカット点を検出するカット点検出過程を具備するものである。

50 【0030】この(1)の手段によれば、圧縮映像を復号化することなく、符号化情報だけを用いてカット点を検出するので、一部の情報を抽出するだけでカット点を検出することができる。

【0031】また、画面間順方向予測符号化画像のみを用いた従来のカット点検出方法では、画面間順方向予測符号化画像と画面内符号化画像の境界でカット点を検出できないという問題点があったが、この(1)の手段では、両符号化方式の画像列から独立に得られた変化量 DP_t 、 DI_t を組み合わせることによって、この問題点を解決している。

【0032】(2)前記(1)のカット点検出方法における画面間順方向予測符号化変化量算出過程において、画面間予測が外れた領域が全画面に占める割合を変化量とするものである。

【0033】この画面間予測符号化方式では、ブロックを符号化する際に、過去の画像と現在の画像の間で類似する領域を検索し予測を行なう。オリジナルの映像に含まれるカメラの動きや被写体の動きは、予測時の類似領域の位置のずれである動きベクトルとして記録され、カメラや被写体の動きがあっても予測が外れることは少ない。

【0034】従って画面間予測が外れた領域が全画面に占める割合を変化量として用いたとき、カメラの動きや被写体の動きは変化量に反映されず、カメラや被写体の動きを誤ってカット点として検出することが少なくなり、前述の問題点(b)を解決することができる。

【0035】(3)前記(1)のカット点検出方法において、画面間順方向予測符号化変化量算出過程、画面内符号化変化量算出過程のいずれか一方又は両方で、画像列の隣合う画像間で変化の生じた領域を変化領域として求める過程と、該変化領域の時間系列に対してあらかじめ与えられた空間的演算を施した後、それらを時間的に積分する過程を含むものである。ここで、「空間的演算を施した後、それらを時間的に積分する」というのは、XOR配列に対応していることである。

【0036】この(3)の手段によれば、画像列の隣合う画像間で変化の生じている領域に対し、空間的演算を施し、フラッシュなどのノイズを除去するので、前述の問題点(c)を解決することができる。また、空間的演算を施した後で時間的積分を行なうことにより、ワイプ等の緩やかな変化を正しく検出することができるので、前述の問題点(d)を解決することができる。

【0037】(4)前記(1)乃至(3)のうちいずれか1つの変化量 DP_t と変化量 DI_t の両方を評価しカット点を検出するカット点検出過程において、画像列 P_t に含まれる符号化情報を基に得られた画像列 P_t の変化量 DP_t に対して、画像列 I_t から得られた情報を用いて、変化量 DP_t を補正する処理過程を含むものである。

【0038】前記(4)の手段によれば、フラッシュ等の瞬間的なノイズの影響を受けづらい大局的な変化特徴量であるIピクチャ変化量を用いてPピクチャ変化量を補正するため、局所的なノイズに左右されず安定にカッ

ト点を検出することができ、前述の問題点(f)を解決することができる。

【0039】(5)前記(1)乃至(3)のうちいずれか1つの変化量 DP_t と変化量 DI_t の両方を評価しカット点を検出するカット点検出過程において、画像列 I_t に含まれる符号化情報を基に得られた画像列 I_t の変化量 DI_t に対して、画像列 P_t から得られた情報を用いて、変化量 DI_t を補正する処理過程を含むものである。

【0040】前記(5)の手段によれば、Pピクチャ変化量を用いるカット点検出方法では、Pピクチャの予測関係が成立しない個所に存在する検出不能なカット点例えば、PピクチャとIピクチャの間のカット点を検出するために、Iピクチャ変化量を用いるが、Iピクチャの変化量は大局的な変化を示しており、そのままではPピクチャとIピクチャにおける変化の有無だけを判定することは不可能なのだが、細部に渡る変化であるPピクチャからの情報を用いてIピクチャ間の変化量を補正することによって、PピクチャとIピクチャの間に発生する変化を検出することができ、前述の問題点(e)を解決することができる。

【0041】(6)前記(1)、(4)、(5)のいずれか一つのカット点検出過程において、画面間順方向予測符号化変化量算出過程から得られる変化量 DP_t が、第一の閾値 TP_t を上回るか、画面内符号化変化量算出過程から得られる変化量 DI_t が、第二の閾値 TI_t を上回るか、どちらか一方が成立した場合に時刻tにカット点が存在していると判定するものである。

【0042】(7)前記(1)、(4)、(5)のいずれか一つのカット点検出過程において、画像列 P_t と I_t の出現頻度を測定する過程を含み、該出現頻度に応じて、二つの変化量 DP_t と DI_t を評価する手続きを切り替えるものである。これにより、画像列 P_t と I_t の出現頻度に応じて評価の方法を使い分けることができる。

【0043】(8)前記(1)、(4)、(5)のいずれか一つのカット点検出過程において、画像列 P_t と I_t の出現頻度を測定する過程を含み、該出現頻度に応じて、二つの変化量 DP_t 、 DI_t に対する二つの閾値 TP_t 、 TI_t を制御するものである。

【0044】前記(6)、(7)及び(8)の手段によれば、カット点の検出に際して、Iピクチャの出現頻度とPピクチャの出現頻度に応じて、Iピクチャ列から求めた変化量 DI_t とPピクチャ列から求めた変化量 DP_t の組み合わせ方を変化させてカット点を検出している。Iピクチャの出現頻度が高い場合にはIピクチャ列から求めた変化量 DI_t を主に用い、変化量 DP_t は検出精度の向上をさせるために、補助的に用いる。逆にPピクチャの出現頻度が高い場合には、変化量 DP_t に重きを置いて評価する。従って、2種類の変化量を各ピクチャの出現頻度に応じて組み合わせてカット点を検出するた

め、各ピクチャの出現頻度によらず高い精度でカット点を検出でき前述の問題点 (g) を解決することができる。

【0045】 (9) 画面間予測符号化方式と画面内符号化方式を共に含む圧縮符号化方式によって圧縮された画像データ列からカット点を検出するカット点検出装置において、画像データ列中の圧縮符号化方式の種類を判定し、符号化データを分類するピクチャ判定手段と、画面間順方向予測符号化方式で圧縮された画像列 P_t に含まれる符号化情報を基に、画像列 P_t の変化量 DP_t を算出するPピクチャ変化量算出手段と、画面内符号化方式で圧縮された画像列 I_t に含まれる符号化情報を基に、画像列 I_t の変化量 DI_t を算出するIピクチャ変化量算出手段と、両変化量算出手段で得られる二つの変化量 DP_t と DI_t を評価しカット点を検出するカット点検出手段とを具備するものである。

【0046】 なお、コンピュータに、前記 (1) 乃至 (8) のいずれか一つのカット点検出方法を機能させるための処理プログラムは、コンピュータが読み取り可能な記録媒体に格納されていても良い。

【0047】 この (9) の手段のカット点検出装置によれば、前記 (1) 乃至 (8) の手段のカット点検出方法を実施することができる。

【0048】

【発明の実施の形態】 以下、図面を参照して本発明の実施の形態 (実施例) を詳細に説明する。

【0049】 (実施形態1) 図1は本発明の実施形態 (実施例) 1のカット点検出方法の処理の流れを示すフローチャートである。本実施形態1のカット点検出方法は、図1に示すように、入力圧縮画像データ0を画面間順方向予測符号化画像列1と画面内符号化画像列4に分離し、それぞれを画面間順方向予測符号化変化量算出過程2と、画面内符号化変化量算出過程5に入力する。各変化量算出過程2及び5において、各画像列の画像の内容が大きく変化したとき大きな値を示すような特徴量を画像列の変化量 (時間的変化量) として算出する。変化量算出過程の実施例については後述する。得られた2つの変化量3及び6を、データ列のカット点検出過程7に入力し、最終的なカット点8を得る。すなわち、2種類の画像列から別々に変化量を算出し、両者を組み合わせて評価することによってカット点を検出する。カット点検出過程の実施例については後述する。

【0050】 図2は本実施形態1のカット点検出方法を実施するカット点検出装置の一実施例の概略構成を示すブロック構成図であり、10は入力圧縮画像データ列、11はピクチャ判定手段、12はデータ列メモリ、13はPピクチャ列、14はPピクチャ変化量算出手段、15はPピクチャの変化量、16はIピクチャ列、17はIピクチャ変化量算出手段、18はIピクチャの変化量、19はカット点検出手段、20はカット点である。

【0051】 本カット点検出装置は、汎用コンピュータとソフトウェアで実現可能な装置であり、図2に示すように、入力圧縮画像データ列10からカット点20を検出する。本カット点検出装置では、入力圧縮画像データ列10は、MPEG方式で圧縮符号化され、Iピクチャが1フレーム、Pピクチャが14フレーム続き、それらを1個のグループとし、そのグループが $t+1$ 個存在する画像データ列 ($IP_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}$) 0, ($IP_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}$) 1, ..., ($IP_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}$) t を扱う。ここで、Iピクチャ列を I_h 、Pピクチャ列を $P_{h,k}$ で表わす。ただし、 $h=0, 1, \dots, t$ 、 $k=0, 1, \dots, 13$ 、 h はグループの番号、 k はグループ内のフレーム番号である。

【0052】 なお、入力圧縮画像データ列10は、ハードディスクやCD-ROM等蓄積装置に保存されている画像ファイルであったり、ネットワークに接続され、伝送される画像ファイルであったりする。サイズ、サンプルレートは任意であるが、画面内符号化方式と画面間予測符号化方式の両方の方式を取り入れてなければならない。例として、H. 261やMPEG1、MPEG2等の形式がある。これらの圧縮符号化方式では、画面内符号化画像、画面間順方向予測符号化画像、画面間双方向予測符号化画像が出現するが、その出現頻度は任意で構わない。

【0053】 入力圧縮画像データ列10を数フレームに渡って格納するデータ列メモリ12が設けられている。本実施例では、データ列メモリ12には、各ピクチャデータと、先頭ピクチャからの通し番号と、ピクチャ判定手段11で判定されるピクチャの種類と、注目している画像がIピクチャの場合は、先頭からのIピクチャ列16の通し番号を、注目している画像がPピクチャの場合は、先頭からのPピクチャ列13の通し番号を保存している。データ列メモリ12は、符号化データの全部乃至一部、画像の先頭ファイルからのフレーム番号や、前処理によって得られる特徴量等の補足情報以外に、外部の機器を用いて入手し、付加される撮影時間、撮影場所等の外部情報を付加してもよい。

【0054】 データ列メモリ12から読み出された画像列のうちPピクチャ列 $P_{h,k}$ ($h=0, 1, \dots, t$ 、 $k=0, 1, \dots, 13$) 13をPピクチャ変化量算出手段14に入力して変化量 $DP_{h,k}$ ($h=0, 1, \dots, t$ 、 $k=0, 1, \dots, 13$) 15を得る。Pピクチャ列13から得られる変化量 ($DP_{h,k}$) 15は、Pピクチャ中のブロックの種類の個数や位置、ブロックに記録されている動きベクトルなど、画像の内容変化に伴って変動する特徴量を用いて算出する。その詳細については後述する。

【0055】 データ列メモリ12から読み出された画像

列の各ピクチャの種別をピクチャ判定手段11で判定し分類されたIピクチャ列 I_h ($h=0, 1, \dots, t$) 16をIピクチャ変化量算出手段17に入力して変化量(DI_h) 18を得る。Iピクチャ列16から得られる変化量(DI_h) 18は、Iピクチャ中のブロックに含まれる輝度、色差の隣接画像間で各画素の差分の総和や、差分の著しい画素の個数や、輝度、色差を基に算出される色相や、それらを基にしたヒストグラム等、画像の内容の変化が反映される特徴量を用いて算出する。その詳細については後述する。

【0056】2つの変化量(DP_h, k) 15と(DI_h) 18をカット点検出手段19にて評価し、カット点20を得ている。2つの変化量(DP_h, k) 15と(DI_h) 18は、Iピクチャ列16、Pピクチャ列13の画像の内容の変化を反映したものである。本実施例では、2つの変化量(DP_h, k) 15と(DI_h) 18のそれぞれから別個にカット点を検出しそれぞれを組み合わせてカット点を検出する例と、2つの変化量(DP_h, k) 15と(DI_h) 18のそれぞれにある重みを乗じて和をとったものからカット点を検出する例の2つの例について後述する。その際の重みや閾値は、あらかじめ与えられているものとしているが、Iピクチャ、Pピクチャのそれぞれの出現頻度に応じて変化させても構わない。変化のさせかたについては後述する。

【0057】以下、Pピクチャ変化量算出手段14の実施例、Iピクチャ変化量算出手段17の実施例及びカット点検出手段19の実施例の詳細な説明を順に行なう。

【0058】Pピクチャ変化量算出手段14の第一の実施例は、Pピクチャに含まれている情報を用いて変化量を算出する。Pピクチャのブロックには予測が成功したブロック（以下pブロック）と、予測が成功しなかったブロック内符号化ブロック（以下iブロック）の2種類が存在する。カット点においては、画面が急激に変化するため、予測がはずれiブロックの個数が急激に増大する。この性質を用いて各フレーム毎のiブロックの個数がブロックに占める割合を各ブロック毎の変化量として算出する。この変化量は、カメラの動きや被写体の動きを誤ってカット点として検出することを防ぐ効果がある。

【0059】通常のカット点ではカット点以降、類似した画像が継続するためPピクチャ中のiブロックの個数は少ない。

【0060】ワイプにはさまざまなタイプがあり、カット点の前後の2枚の静止画像の前面の画像の一部が徐々にめくれていくようなワイプの場合、めくれていく部分に沿ってiブロックが出現してくる。ワイプ中の各Pピクチャに出現するiブロックはごく少数であるが、数フレームに渡るiブロックの個数を累積したものは大きい。

【0061】フラッシュは1フレームだけ急激に明度が

上昇し直後に下降するため、フラッシュがたかれたPピクチャとその直後のPピクチャでは、ほぼ同じ位置にiブロックが出現する。

【0062】本第一の実施例における通常のカット点では、変化量は十分に大きな値となる。しかし、フラッシュもカット点と同様大きな変化量を示したり、緩やかに変化するカット点では、変化量が大きくならないという欠点を持っている。

【0063】次に、Pピクチャ変化量算出手段14の第二の実施例を、図3を用いて説明する。まず、Pピクチャ列(P_h, k) 110が入力される。入力された各Pピクチャに対し、各Pピクチャのブロック情報読み取り処理100により、Pピクチャ中のブロックの位置と種別を読み取る。これによりPピクチャのブロック配列111を得る。

【0064】ピクチャのブロック配列111のブロック配列のiブロックを“1”pブロックを“0”とみなし、XOR配列作成処理101では、図4に示すように、注目Pピクチャと前XOR配列 $X_{h, k-1}$ との間でXORをとり、注目PピクチャのXOR配列 $X_{h, k}$ (112)を作成する。XOR配列112の作成を一定回数繰り返したら、XOR配列をリセットする。XOR配列をリセットする周期は任意に設定できる。周期を長くすると緩やかな変化は検出しやすくなる反面、短い間隔で発生するカット点を検出することができない。

【0065】本第二の実施例ではIピクチャから次のIピクチャまでの14フレームでXOR配列を順次作成する。一般的なカット点、ワイプ、フラッシュをそれぞれ含むPピクチャ列に対するXOR配列の変化の様子を図5乃至図7に示す。一般的なカット点(図5)では、カット点の直後に多数のiブロックが発生するが、その後には表われるiブロックの個数は少ないため、カット点の後のXOR配列上では1の個数は高い値を保ったままである。ワイプの場合(図6)、iブロックの位置が少しずつずれて出現するため、XOR配列上で1が蓄積されていく。フラッシュの場合(図7)、フラッシュと直後のPピクチャで、iブロックがほぼ同じ位置にほぼ同じ個数だけ出現するため、XOR配列上で一時的に1の個数が増大するもののすぐに打ち消されてなくなる。

【0066】変化量算出処理102の処理手順を図8を用いて説明する。XOR配列 $X_{h, k}$ に含まれる1の個数をカウントしたものを $N_{h, k}$ として出力する(ステップ1031)。グループ内にカット点が存在する場合、グループの最終XOR配列 $X_{h, 13}$ の1の個数 $N_{h, 13}$ は高い値を持つので、 $N_{h, 13}$ が一定の閾値を超えた(Yes)場合(ステップ1032)、カット点の候補がグループhに存在するとし、カット点候補のフレーム番号を特定するために、過去に遡って $N_{h, k}$ を調べ、その値がある閾値(前記一定の閾値と同じである)を超え始めたところをカット点候補のフレーム番号 k' とする(ステ

ップ1033)。特定されたフレーム番号に対応する $N_{h,k'}$ がブロックの総数に占める割合を変化量 $15D_{Ph,k'}$ として算出し、他は0とする。カット点が存在しないと判定された(No)場合、変化量は、すべて0とする(ステップ1034)。

【0067】Pピクチャ変化量算出手段の第二の実施例では、iブロックの個数を基に、容易にカット点を検出しやすいよう、通常のカット点やワイプ、ディゾルブ等の緩やかなカット点を強調し、明らかにカット点ではないと思われるノイズを除去する処理を行なって変化量を算出した。通常のカット点ではカット点以降、類似した画像が継続するためPピクチャ中のiブロックの個数は少ない。

【0068】ワイプにはさまざまなタイプがあり、カット点の前後の2枚の静止画像の前面の画像の一部が徐々にめくれていくようなワイプの場合、めくれていく部分に沿ってiブロックが出現してくる。ワイプ中の各Pピクチャに出現するiブロックはごく少数であるが、数フレームに渡るiブロックの個数を累積したものは大きい。

【0069】フラッシュは1フレームだけ急激に明度が上昇し直後に下降するため、フラッシュがたかれたPピクチャとその直後のPピクチャでは、ほぼ同じ位置にiブロックが出現する。

【0070】第二の実施例では、カット点を強調し、ノイズを除去するために、XOR配列を用いているが、ヒストグラムを作成して変化量を算出したり、連続する数フレームの平均値を基に変化量を算出したりするなどの空間的演算を行ってもよい。

【0071】次に、Iピクチャ変化量算出手段17について説明する。Iピクチャ変化量算出手段17は、Iピクチャ列16から変化量を算出する。Iピクチャにはiブロックしか存在しない。iブロック中のDC-T係数の直流成分は、各iブロック中の画素の輝度や色差成分の平均値を表わしているため、直流成分から縮小画像を生成できる。本実施例では、直流成分のうちの輝度成分を用いて縮小画像を作成し、隣接縮小画像の各画素同士の差の大きい画素の個数を変化量とする。なお、隣接縮小画像の画素同士の比較において、比較対象は必ずしも対応する位置に存在する画素でなくとも良く、対応する位置周辺に一つでも類似する画素が存在するかどうかで変化量を算出しても良い。これは、カメラワーク等の動き影響を排除するために有効な手法である。なお、探索範囲については、Iピクチャ同士の間に存在するピクチャ数に応じて、範囲を決定する。Iピクチャ間にピクチャが多数存在する場合には、探索範囲を広げ、少数しかピクチャが存在しない場合には、探索範囲を狭めるという手法をとる。また、Pピクチャに記録されている動き補償によって得られている動きベクトルを元に、対応する画素がどのような軌跡を描いたのかを追跡するという処

理を行っても良い。

【0072】次に、図9を用いてIピクチャ変化量算出手段の第一の実施例の詳細な説明を行なう。Iピクチャ列16から直流成分抽出処理200によってiブロックの直流成分の輝度成分から直流成分輝度画像 DC_h ($h=0, 1, \dots, t$) 210を得る。Iピクチャ直流成分同士の比較処理201で、隣り合う直流成分画像の輝度の差分があらかじめ与えられた閾値 T_h より大きい差分画素数(N_h) 211を算出する。画像の内容が類似していれば、差分画素数(N_h) 211は小さく、違いが大きければ差分画素数(N_h) 211は大きくなる。

【0073】ただし、カメラや被写体の動きによっても差が生じるので、突出検出フィルタ処理202によって、こうしたノイズを消去する。突出検出フィルタ処理202は、注目している特徴量 N_h と、前後数フレームの特徴量 $\dots, N_{h-2}, N_{h-1}, N_{h+1}, N_{h+2}, \dots$ のうちの最大のものの差をとるものである。

【0074】本実施例では前後2フレームの特徴量を比較対象に用いている。突出検出フィルタ処理202によってカメラの動きや、被写体の動き等のノイズは除去し、カット点を強調した変化量 $18DI_h = N_h - \max\{N_{h-2}, N_{h-1}, N_{h+1}, N_{h+2}\}$ を得る。

【0075】変化量算出の際に作成する縮小画像は、直流成分のうちの輝度成分だけを用いた白黒画像でも良いし、色差成分も用いてカラー画像化しても良い。変化量は、隣り合うIピクチャ同士の変化の大小を反映する特徴量であれば良く、例えば、対応する画素同士の輝度や色の差分でも良いし、各画像の輝度と色を基に算出したヒストグラム同士の差分でも良いし、いくつかの画素を集めた画素ブロック同士の差分をとっても良い。本実施例では、変化量の算出に際して、突出検出フィルタをかけたが、他のフィルタを用いても良いし、フィルタを用いずとも良い。

【0076】カット点検出過程の一実施例について述べる。この過程では、Iピクチャ変化量算出手段17と、Pピクチャ変化量算出手段14の2つの変化量算出手段によって得られたIピクチャ変化量 DI_h とPピクチャ変化量 $DP_{h,k}$ を組み合わせることで、最終的なカット点を得る。

【0077】次に、カット点検出手段19の第一の実施例について述べる。2つのIピクチャ変化量 DI_h ($h=0, 1, \dots, t$)とPピクチャ変化量 $DP_{h,k}$ ($h=0, 1, \dots, t, k=0, 1, \dots, 13$)に対して、それぞれ閾値 T_P と T_I を設定し、各変化量が閾値を上回った場合、全てをカット点とする。なお、ここで用いる閾値は、あらかじめ定めた値でも構わないし、様々に変化させても構わない。変化のさせ方については後述する。

【0078】前記カット点検出手段19の第一の実施例について、図10を用いて詳細に説明する。カット点が

存在するかどうかの判定をステップ300でグループ毎に行なう。あるグループ h ($h=0, 1, \dots, t$) のPピクチャ列13から算出したPピクチャ変化量 DP_h, k ($k=0, 1, \dots, 13$) が閾値 TP を上回るPピクチャ変化量 DP_h, k' が存在する (Yes) 場合、フレーム番号 k' にカット点 (カット点 k' : 302) が存在するとし、Iピクチャ変化量 DI_h の値とは無関係にグループ h に対するカット点検出処理を終了する。

【0079】Pピクチャ変化量 DP_h, k が閾値 TP を下回る場合でも、グループの最後のPピクチャと次のグループの先頭のIピクチャの間にカット点が存在する可能性があるため、Iピクチャから算出された DI_{h+1} を評価する (ステップ303)。 DI_{h+1} が閾値 TI を上回る場合 (Yes)、グループ h の最後のPピクチャと次のグループ $h+1$ の先頭のIピクチャとの間にカット点が存在するとし、カット点に対応するフレームの位置を決定し、カット点 $h+1$ に対応する時刻をカット点 (カット点 $h+1$: 305) として出力して終了する。 DI_{h+1} が閾値 TI を下回った (No) 場合には、カット点はグループ h には存在しないとし (ステップ306)、終了する。

【0080】次に、カット点検出手段19の第二の実施例について説明する。第二の実施例では、2つの変化量算出手段 (14, 17) で得られた2つの変化量に重みを乗じ、和をとったものが閾値以上である場合カット点が存在するとする。

【0081】前記カット点検出手段19の第二の実施例について、図11を用いて詳細に説明する。まず、ステップ400であるグループ h のPピクチャ列13から算出した変化量の最大値 $\max \{DP_h, k\}$ ($k=0, 1, 2, \dots, 13$) を決定し、前記グループ h にカット点が存在するかどうかの判定をステップ401で行なう。あるグループ h のPピクチャ列13から算出した変化量の最大値 $\max \{DP_h, k\}$ と、次のグループ $h+1$ のIピクチャ列16の変化量 DI_{h+1} のそれぞれに重み WP, WI (ただし、 $WP=14/15, WI=1/15$) を乗じ和をとる。得られた値が閾値 T を上回る (Yes) 場合、グループ h にカット点は存在しているとする (ステップ402)。存在すると判定された場合、フレーム番号を特定するため、重みを乗じた変化量 $WP * \max \{DP_h, k\}$ と $WI * DI_{h+1}$ とをステップ403で比較し、変化量 $WP * \max \{DP_h, k\}$ が大きい (Yes) 場合、 $\max \{DP_h, k\}$ に対応するフレーム番号 k' にカット点が存在すると判定する (ステップ405)。また、 $WI * DI_{h+1}$ が大きい (No) 場合、 $WI * DI_{h+1}$ の直前にカット点が存在と判定する (ステップ406)。

【0082】また、ステップ401で前記グループ h にカット点が存在しないと判定された (No) 場合、カット点は存在しないと処理は終了する (ステップ40

4)。

【0083】なお、ここで用いる重みと閾値は、あらかじめ定められた値でも構わないし、様々に変化させても構わない。変化のさせかたについては後述する。

【0084】次に、前記カット点検出手段19の第3の実施例について説明する。前述のカット点検出手段の第一の実施例における閾値を出現頻度に応じて変化させる。Iピクチャの出現頻度に応じた最適な閾値を与える対応表をあらかじめ作成しておき、出現頻度に応じて閾値を設定する。Iピクチャの出現頻度が高い場合には、Iピクチャ同士の間隔は接近しているため、カメラや被写体の動きによる画像の内容の違いは小さく、変化量は小さいため、閾値を低くしてもこれらを実点として誤って検出することはない。閾値を低くすることで、シーンの画像の内容の変化が小さいカット点を検出し損なうことがなくなる。

【0085】Iピクチャの出現頻度が低い場合は、カメラや被写体の動きによる差が大きく、変化量が大きくなるため、これらを実点として誤って検出することを防ぐため閾値を高く設定する。Iピクチャの出現頻度が非常に高い場合には、Iピクチャの変化量だけを用いてカット点を検出する。IピクチャからDCT係数の直流成分を抽出すると、各Iピクチャの縮小画像を作成することができる。この縮小画像を従来の非圧縮映像からのカット点検出方法に入力することで、従来のカット点検出方法とほぼ等しい精度でカット点を検出することができる。本実施例では、Iピクチャが1秒間に10フレームを超えて出現する場合は、Iピクチャのみでカット点の検出を行なうことにしている。

【0086】Iピクチャの閾値 TI と同様に、Pピクチャ閾値 TP を変化させる。Pピクチャは動き予測符号化を行っているため、動きの影響を受けづらいという利点があり、IピクチャないしPピクチャが連続して出現している場合には、とくにPピクチャ閾値 TP を変化させる必要はない。しかし、Pピクチャ列の間にBピクチャが出現する場合には、ピクチャ同士の間隔が広い場合、画像中の被写体やカメラの動きが連続に出現する場合と比べて大きく変化する。従って、Pピクチャが連続で出現しているか、それともBピクチャが出現しているかどうかによって、Pピクチャ閾値 TP を変化させる。すなわち、Bピクチャの出現頻度に応じてPピクチャ変化量の閾値 TP を変化させる。

【0087】例えば、Pピクチャ2フレームの間にBピクチャが0フレームも出現しない場合と、Bピクチャが2フレーム出現する場合とでは、閾値を変化させる。動きの影響を受けやすいとイントラブロックが多数出現しやすく、変化量も大きくなりやすいという特徴があるため、Bピクチャが0フレームも出現しない場合は閾値を低くし、2フレーム出現する場合には、閾値を高くするという操作を行なう。

【0088】また、Iピクチャ、Pピクチャのそれぞれの変化量への重みの付け方についても、双方の出現頻度に応じて変化させる。すなわち、後述する第二の実施例では、WIを1/15、WPを14/15としているが、Iピクチャの出現頻度が高ければWIは大きくし、低ければWIを小さくする。また、Bピクチャの出現頻度が高ければWPは大きくし、低ければWPを小さくする。

【0089】なお、第三の実施例同様Iピクチャが1秒間に10フレーム以上出現するような場合には、WIを15/15、すなわち1としWPを0とする。

【0090】本実施例では、閾値はカット点検出装置によって自動的に与えられているが、ユーザインタフェース部を付加し、任意に閾値を変化することも可能である。例えば、ユーザが圧縮符号化映像の内容をブラウジングしやすいようにカット点直後の画像を表示する機能と、ユーザインタフェース部とを共に付加し、閾値を任意に設定させることで、ユーザが圧縮符号化映像の内容を表示する画像の個数を調節し、希望するシーンをアクセスしやすいようにする本発明の装置を応用したアプリケーションも考えられる。

【0091】(実施形態2) 図12は本発明の実施形態(実施例)2のカット点検出方法を説明するための横切り物体が存在する場合の変化量を示す図、図13はIピクチャ上の変化領域部分によるDP算出処理を説明するための図、図14は各状態におけるPピクチャの変化量を説明するための図である。

【0092】本発明の実施形態(実施例)2のカット点検出方法は、前記実施形態1におけるPピクチャ変化量をIピクチャ変化量で補正することにより、ノイズを除去するものである。

【0093】前記実施形態1において、Pピクチャ列から変化量を算出する際に、Pピクチャ上での、フラッシュや動き情報などの急激な変化をカット点として誤って検出することを防ぐ変化量算出手段を説明した。この変化量算出手段の一実施例は、XOR配列を作成する処理から変化量を算出する手法である。これは、フラッシュが発生したフレームと次のフレームでは、それぞれ対応する位置上のブロック(MB)がブロック内符号化ブロック(IntraMB)になるためXOR演算を行なうことで、フラッシュを変化量から除去することが可能である。

【0094】XOR配列による方法は、フラッシュ等の影響を相殺し、ディゾルブ等の緩やかな変化にも有効な手法である。しかし、物体が高速で横切っていく場合、横切る物体の縁領域に追従してブロック内符号化ブロック(IntraMB)が出現する。こうした場合、前記実施形態1ですでに提案したXOR配列を作成する手法では、ノイズが相殺されず、大きな変化量が算出されることになる(図12参照)。一方で、すばやく横切った

め、対応するIピクチャ同士の成分は一致し、Iピクチャ同士の比較による変化は小さい。したがって、Pピクチャ変化量は大きく、Iピクチャ変化量は小さくなり、評価方法によっては、実際にはカット点が存在しなくても、カット点が存在すると誤って判別する可能性がある。

【0095】そこで、これを防ぐために、算出されたPピクチャ変化量から、Iピクチャ同士のブロック(MB)を比較して類似していると判定されたブロック(MB)に対応するPピクチャ列上でのブロック(MB)部分から算出されたPピクチャ変化量部分を除去し、得られる値をPピクチャ変化量として用いる(図13参照)。

【0096】このような場合、Pピクチャ列上に、カット点、ディゾルブ変化、フラッシュ、横切り物体、カメラワーク等による動き情報が存在するとき、それぞれの画像上では(図14参照)、

カット点及びディゾルブ：Iピクチャ同士に著しい差があり、Pピクチャ上での対象領域は大きくなり、Pピクチャ変化量は大きな値を持つ。

フラッシュ及び横切り物体：Iピクチャ同士に差がないため、Pピクチャ上での比較対象領域は小さくなり、Pピクチャ変化量も大きな値を取らない。

動き情報：Iピクチャ同士に著しい差があり、Pピクチャ上での比較対象領域は大きくなるが、Pピクチャ上で、動き補償により動き情報が吸収されるため、Pピクチャ変化量は小さくなる。

すなわち、Pピクチャを算出する際にIピクチャで得られる情報を用いることにより、安定に検出することができ。

【0097】(実施形態3) Iピクチャが周期的に出現するような大半のMPEGデータでは、Pピクチャ変化量だけを基にしたカット点検出方法の場合、Iピクチャ直前にカット点が存在していても検出が不可能であることは、すでに前記実施形態において述べた。そこで、Iピクチャ同士を比較して、Iピクチャ変化量を用いてカット点を検出する方法についてもすでに述べた。

【0098】しかし、Iピクチャ変化量は大局的な特徴量であるため、Iピクチャ直前に存在するカット点を検出するためには、Iピクチャ間に存在するPピクチャ上での変化による影響を排除する必要がある。なお、この場合のPピクチャ上での変化とは、カット点、フラッシュ、カメラワークや動き被写体等の動き情報のことを指す。そこで、Iピクチャ同士の比較によるカット点検出に影響を及ぼしそうなPピクチャ上での変化を記録し、Iピクチャ変化量を算出した後でPピクチャの変化分を除去する補正処理を行なって、Iピクチャ直前のカット点を検出することができ。

【0099】以下に本実施形態3の2つの実施例について説明する。第一の実施例は、単純なIピクチャ同士を

比較して変化量 $D I_t$ を算出すると、カメラワークや動き被写体などの動き情報に左右される。そこで、あらかじめ画面上で動き物体がどのように動いていったかという軌跡を各 P ピクチャの動き予測ベクトルを利用して追跡し、注目 I ピクチャ上の対応するマクロブロックが比較対象 I ピクチャ上のどこに有りそうかという情報を用いて、比較するブロック (MB) の探索範囲を狭め、動きに追従した I ピクチャ同士からの変化量を算出できるようにする。このように、動き情報を相殺することによってカメラワークや動き被写体の影響を排除でき、これらの場合を誤ってシーンチェンジが行われたと判断することを防ぐ。

【0100】第二の実施例は、I ピクチャ列からの変化量は、P ピクチャ変化量では検出不可能な P ピクチャと I ピクチャ間に存在するカット点の検出にも用いられる。その際、単純な I ピクチャ同士の比較による I ピクチャ変化量では P ピクチャ上で発生した変化なのか、P ピクチャと I ピクチャ間で発生した変化なのかを区別できない。そこで、I ピクチャ変化量から P ピクチャ上の変化による影響を除去する処理が必要となる。

【0101】P ピクチャ上での動きや明度変化は、予測残差の大きな $InterMB$ (フィールド間順方向予測符号化ブロック) や $IntraMB$ となって現れる。そこで、P ピクチャ列上で $IntraMB$ や予測残差の大きな $InterMB$ が出現した領域を順次蓄積し、記録する。P ピクチャ上に蓄積し、記録されたこれらの領域は、P ピクチャ上の変化に対応し、I ピクチャ同士の比較を行なって P ピクチャと I ピクチャ間での変化量を算出する際には、除去されるべきである。そこで、算出された I ピクチャ変化量から P ピクチャ上で記録した前述の変化領域に対応した領域から算出された変化量については、除去する処理を行なうことにより、P ピクチャと I ピクチャ間の変化量を求めることができる (図 15 参照)。

【0102】(実施形態 4) なお、本発明は、圧縮符号化データを保存し、それらを自由に読み出し可能なハードディスクやそれに準ずる装置と、データを処理する際に必要なバッファやそれに準ずる装置と、最終的に検出されたカット点を表示、出力するディスプレイなどの装置を備え、それらハードディスク、バッファ及びディスプレイ等をあらかじめ定められた手順に基づいて制御する中央演算装置などを備えたコンピュータやそれに準ずる装置を元に、上述した実施形態 1, 2, 3 の処理、ないしは、図 1, 図 2, 図 3, 図 8, 図 9, 図 10 及び図 11 に示した方法ないしアルゴリズムを記述した処理プログラムやそれに準ずる物を、該コンピュータに対して与え、制御、実行させることで実現することが可能である。なお、該処理プログラムやそれに準ずる物を、コンピュータが実行するに当たって読み出すことを可能とするフロッピーディスク (FD)、光磁気ディスク (M

O) やそれに準ずる記憶媒体に格納されていても構わない。

【0103】以上、本発明者によってなされた発明を、前記実施例に基づき具体的に説明したが、本発明は、前記実施例に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは勿論である。

【0104】

【発明の効果】本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば、以下のとおりである。

【0105】圧縮符号化された画像データ列に対し、画面内符号化画像と、画面間順方向予測符号化画像に含まれている各種情報を用いて、カット点を検出することにより、データを復号化することなく、高速で、安定にカット点を検出することができる。

【図面の簡単な説明】

【図 1】本発明の実施形態のカット点検出方法の処理の流れを示すフローチャートである。

20 【図 2】本実施形態のカット点検出方法を実施するカット点検出装置の一実施例の概略構成を示すブロック構成図である。

【図 3】本実施形態のカット点検出方法を実施するカット点検出装置における P ピクチャ変化量算出手段の処理手段を示すフローチャートである。

【図 4】本実施形態のカット点検出方法における XOR 配列作成の原理を説明するための図である。

30 【図 5】本実施形態のカット点検出方法の一般的なカット点の場合の諸条件における P ピクチャ、ブロック配列、XOR 配列の模式図である。

【図 6】本実施形態のカット点検出方法のワイプの場合の諸条件における P ピクチャ、ブロック配列、XOR 配列の模式図である。

【図 7】本実施形態のカット点検出方法のフラッシュの場合の諸条件における P ピクチャ、ブロック配列、XOR 配列の模式図である。

【図 8】本実施形態のカット点検出方法における変化量算出処理の手順を示すフローチャートである。

40 【図 9】本実施形態のカット点検出方法を実施するカット点検出装置における I ピクチャ変化量算出手段の処理手段を示すフローチャートである。

【図 10】本実施形態のカット点検出方法を実施するカット点検出装置におけるカット点検出手段の第一の実施例の処理手順を示すフローチャートである。

【図 11】本実施形態のカット点検出方法を実施するカット点検出装置におけるカット点検出手段の第二の実施例の処理手順を示すフローチャートである。

【図 12】本発明の実施形態 2 における横切り物体が存在する場合の変化量を説明するための図である。

50 【図 13】本実施形態 2 における I ピクチャ上の変化傾

域部分によるDP算出処理を説明するための図である。

【図14】本実施形態2における各状態におけるPピクチャの変化量を説明するための図である。

【図15】本発明の実施形態3におけるDIのDPによる補正及び比較処理を説明するための図である。

【図16】従来の画面内符号化方式、画面間順方向予測符号化方式、画面間双方向予測符号化方式を説明するための図である。

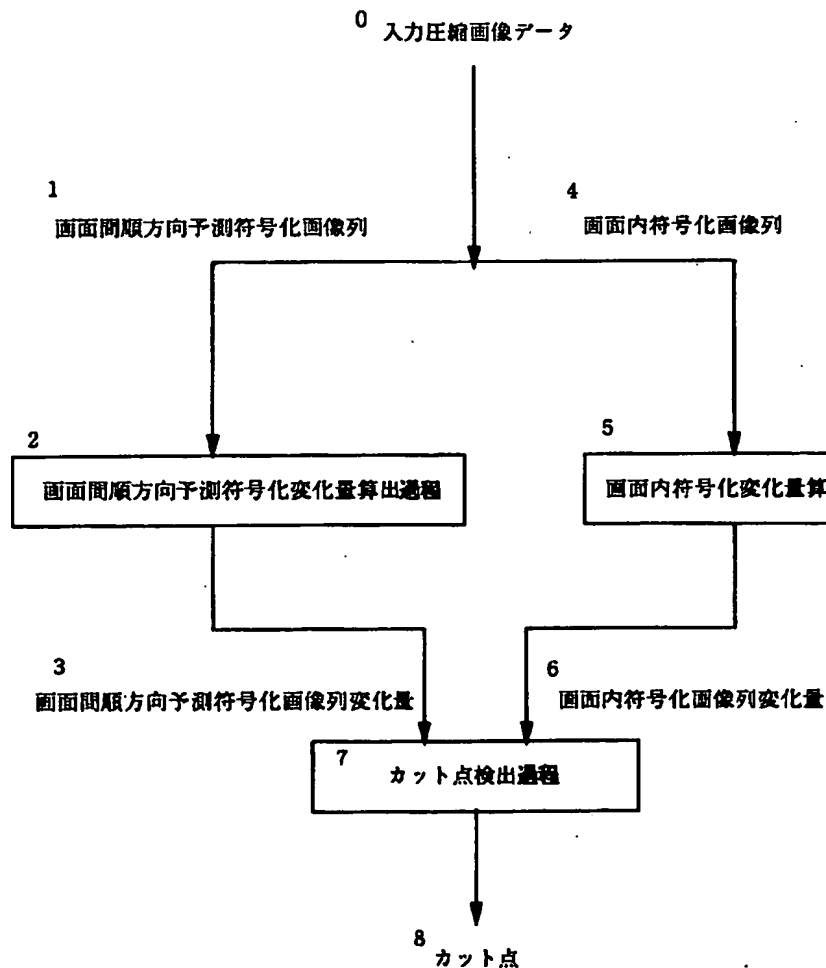
【符号の説明】

0…入力圧縮画像データ、1…画面間順方向予測符号化 10

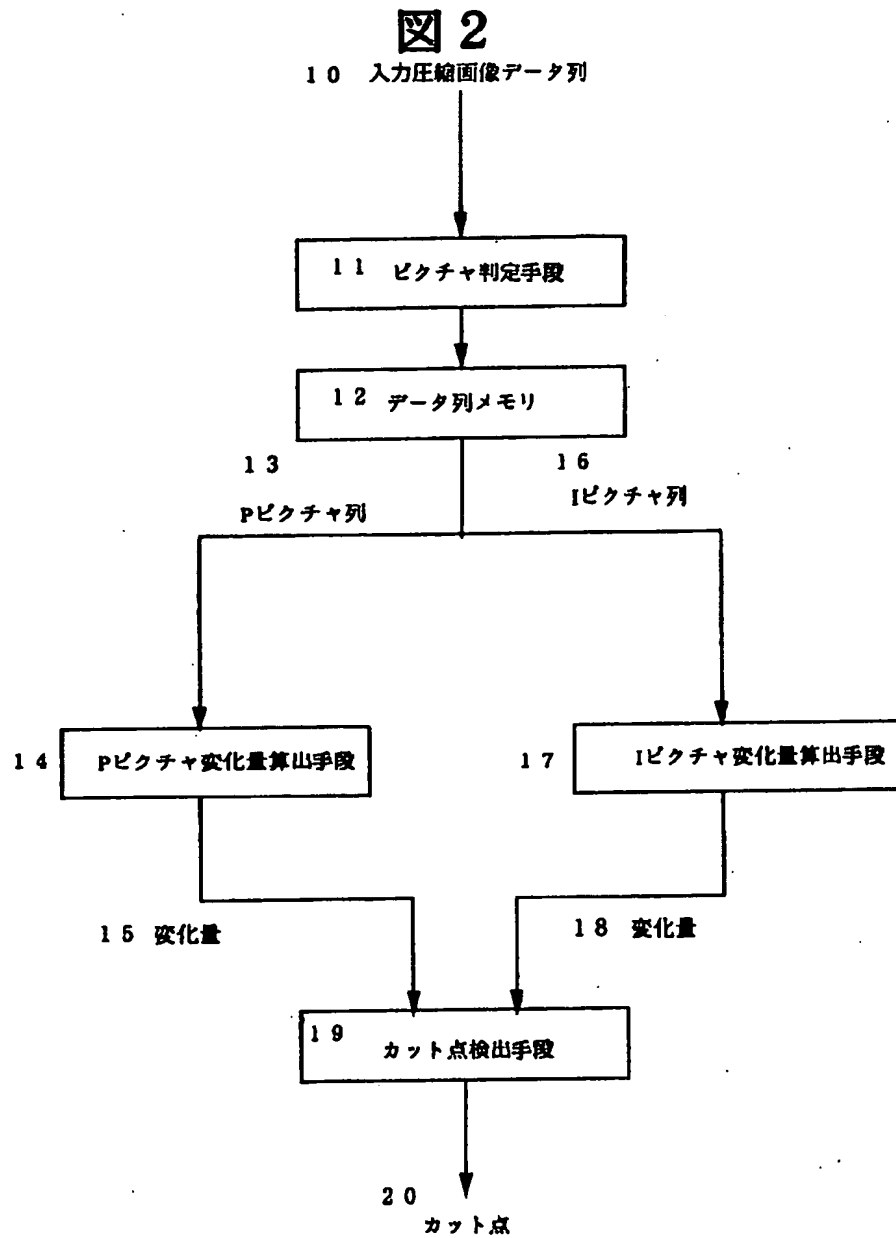
画像列、2…画面間順方向予測符号化変化量算出過程、3…画面間順方向予測符号化画像列変化量、4…画面内符号化画像列、5…画面内符号化変化量算出過程、6…画面内符号化画像列変化量、7…カット点検出過程、10…入力圧縮画像データ列、11…ピクチャ判定手段、12…データ列メモリ、13…Pピクチャ列、14…Pピクチャ変化量算出手段、15…Pピクチャの変化量、16…Iピクチャ列、17…Iピクチャ変化量算出手段、18…Iピクチャの変化量、19…カット点検出手段、20…カット点。

【図1】

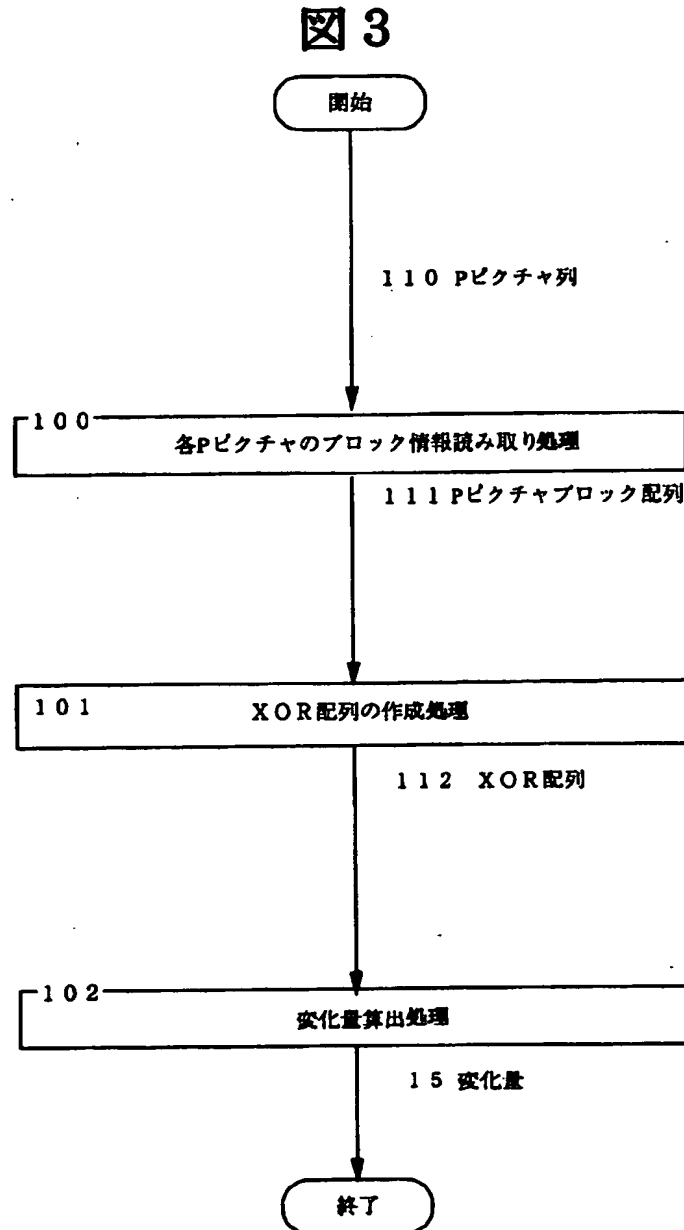
図 1



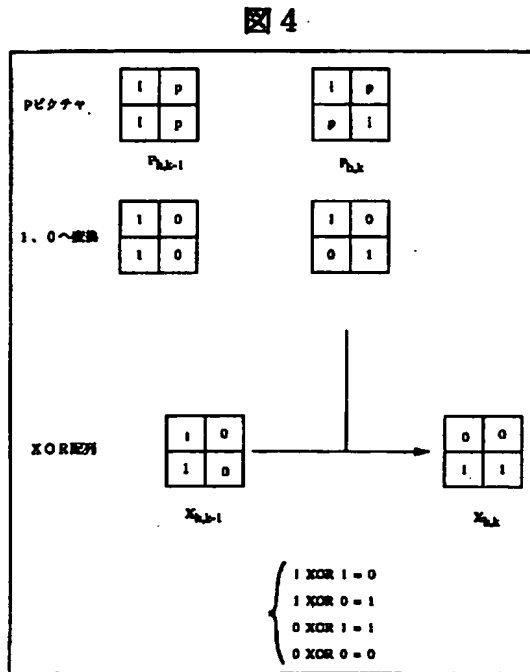
【図2】



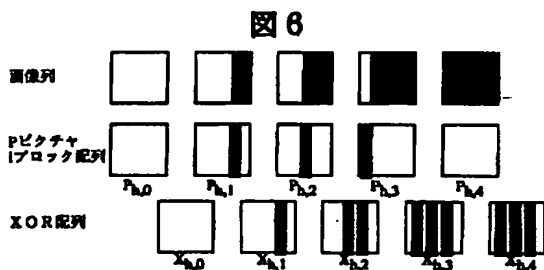
【図3】



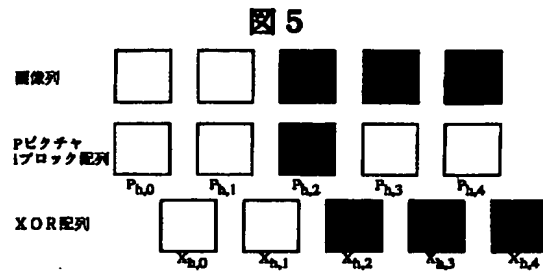
【図4】



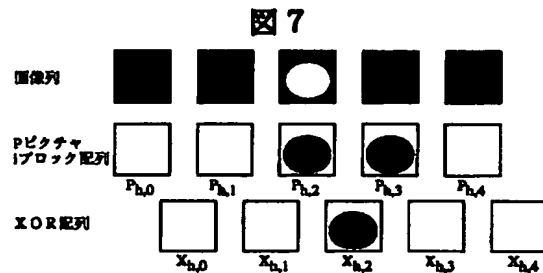
【図6】



【図5】

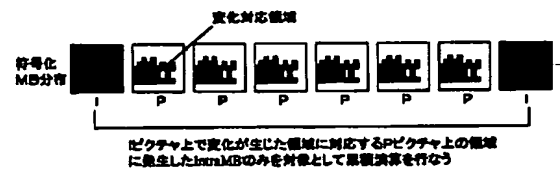


【図7】



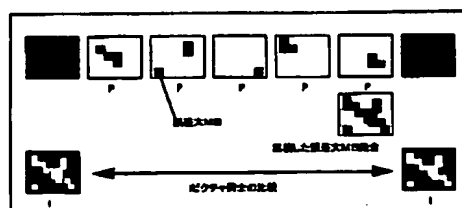
【図13】

図 13

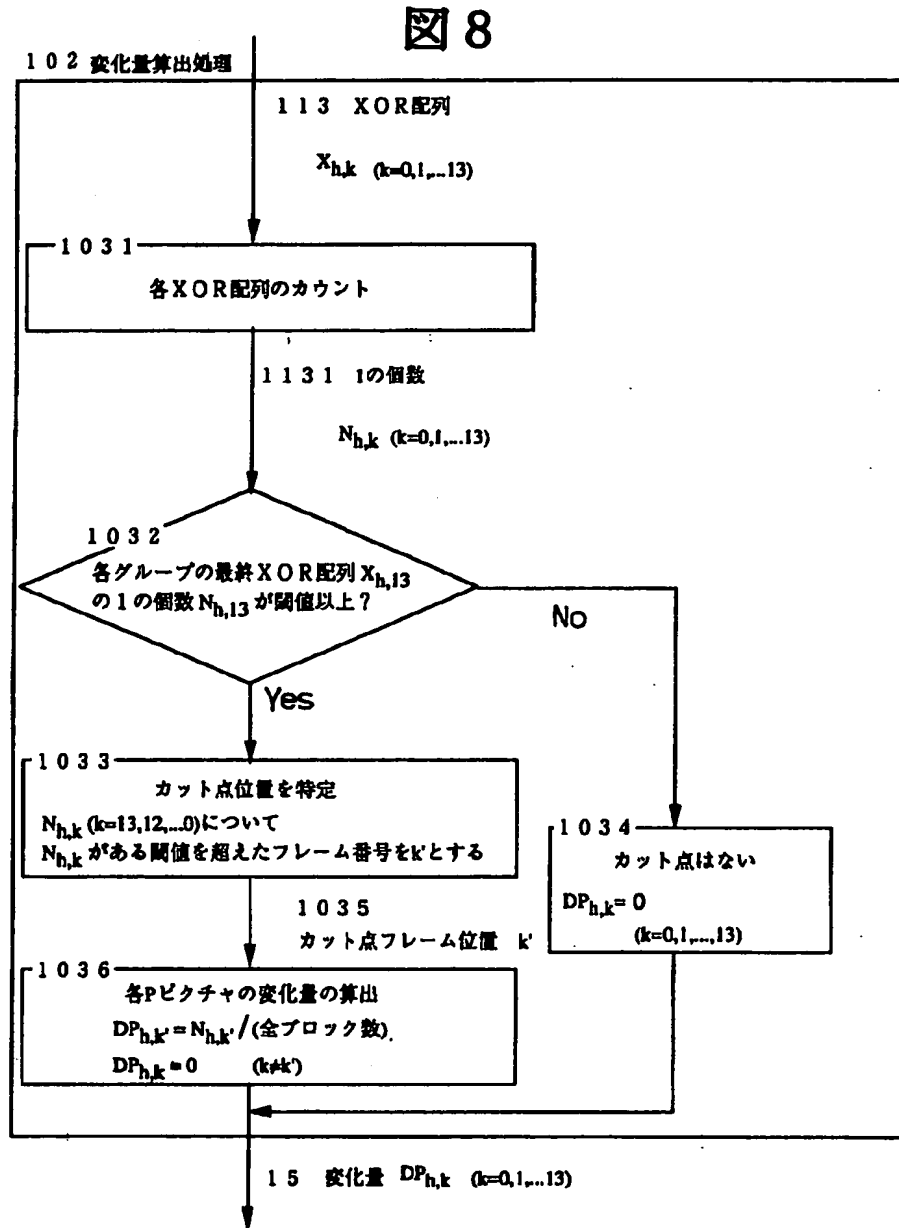


【図15】

図 15

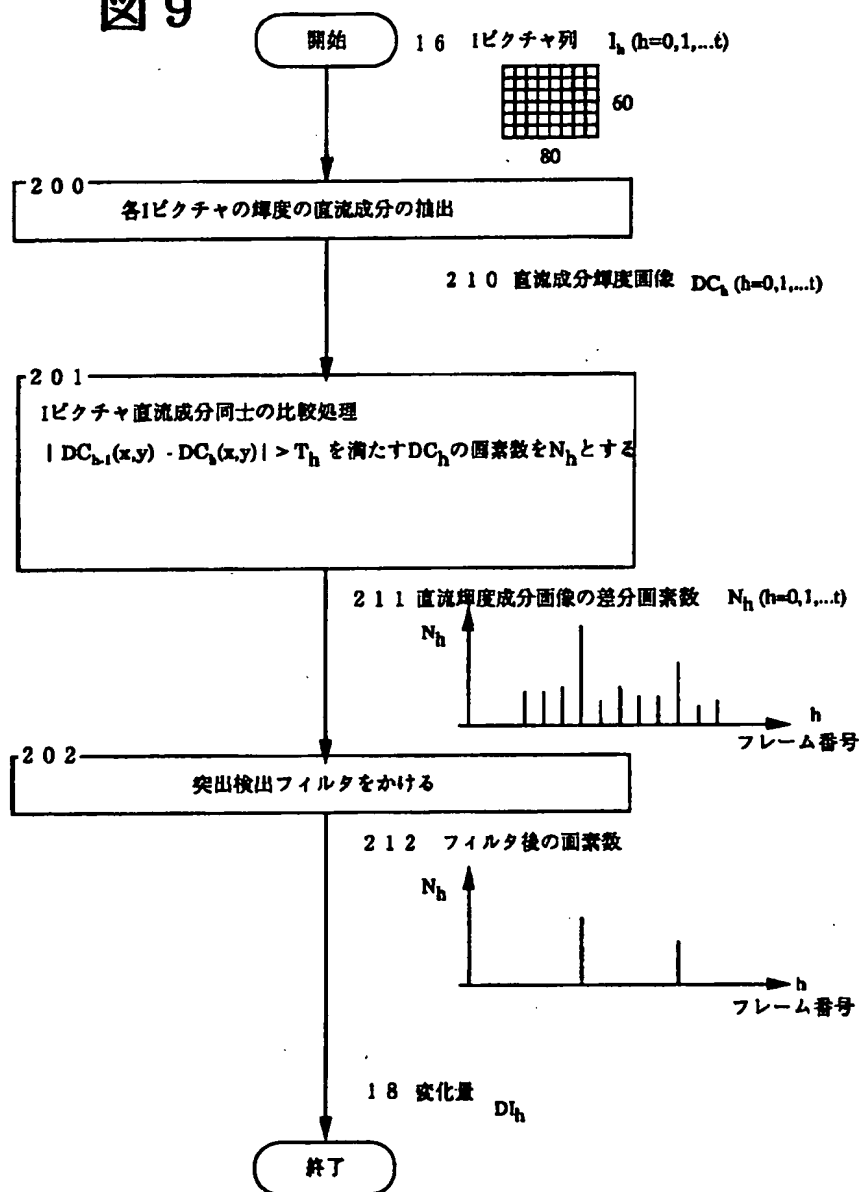


【図8】



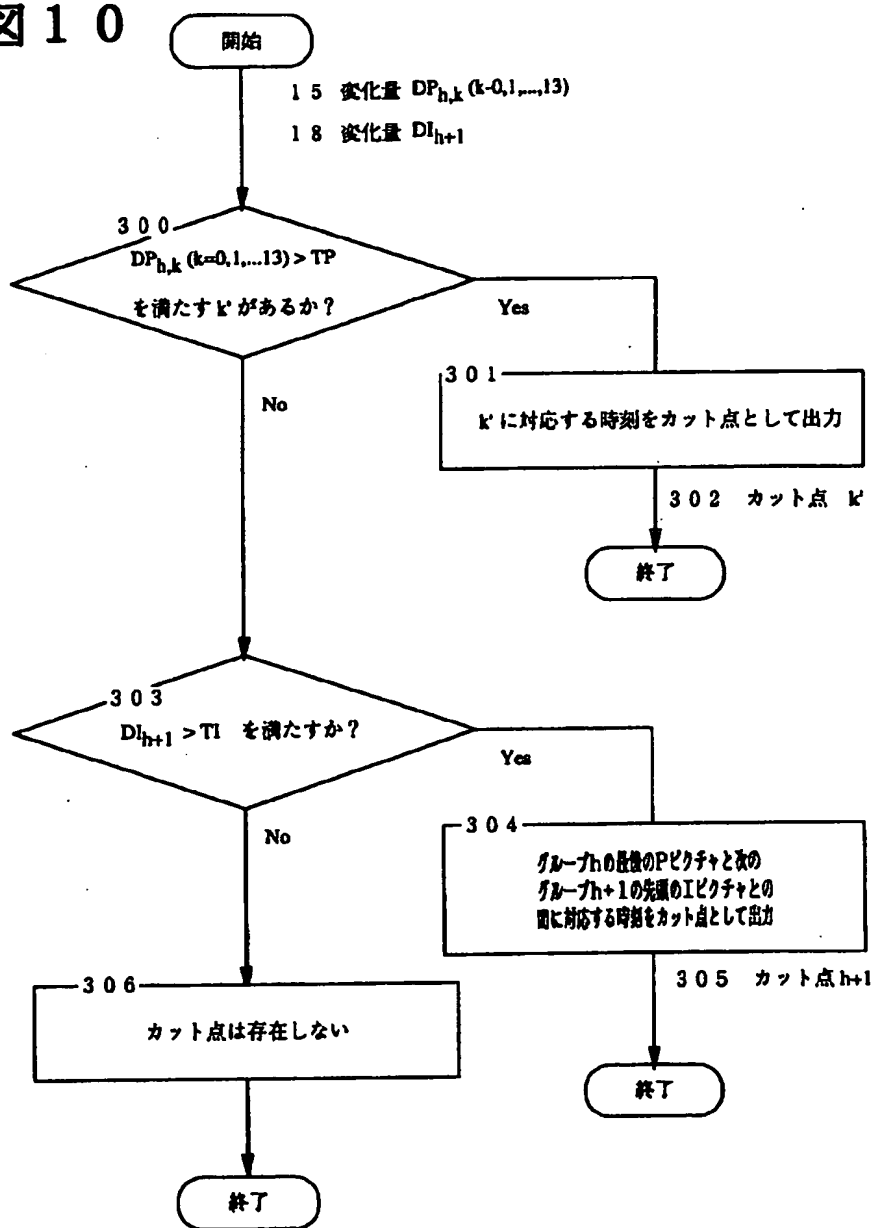
【図9】

図 9



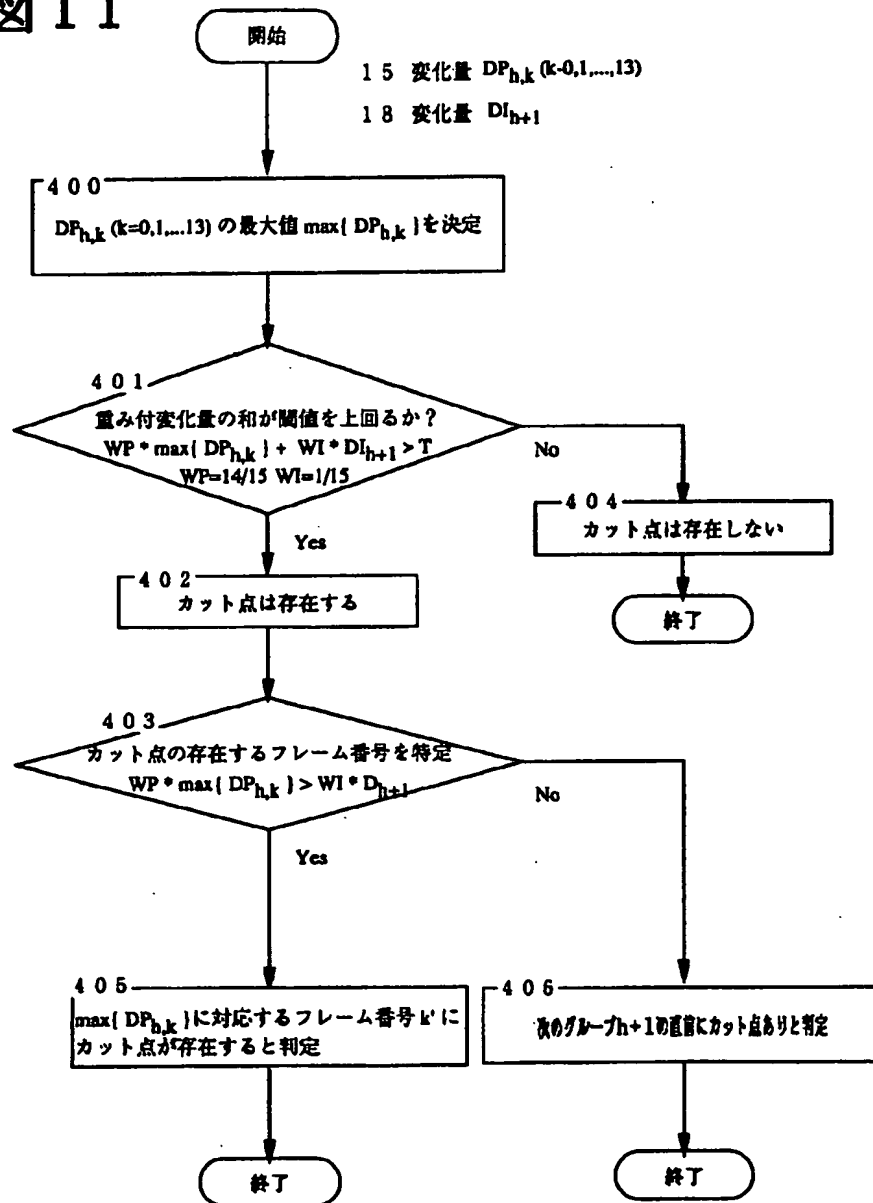
【図 10】

図 10



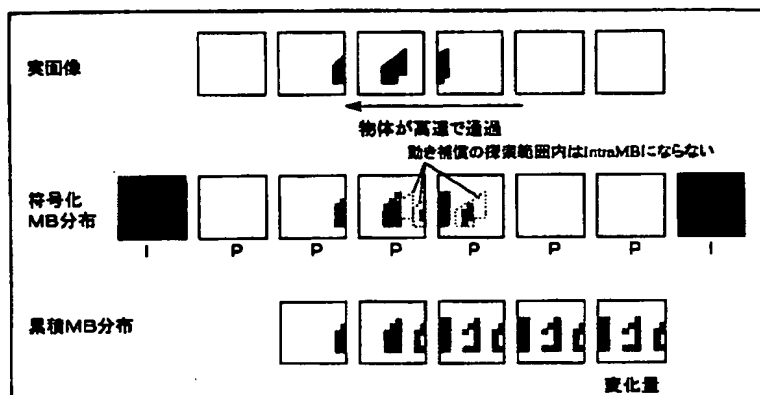
【図 11】

図 11

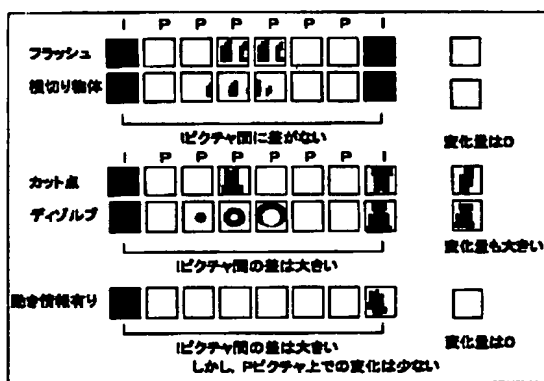


【图 12】

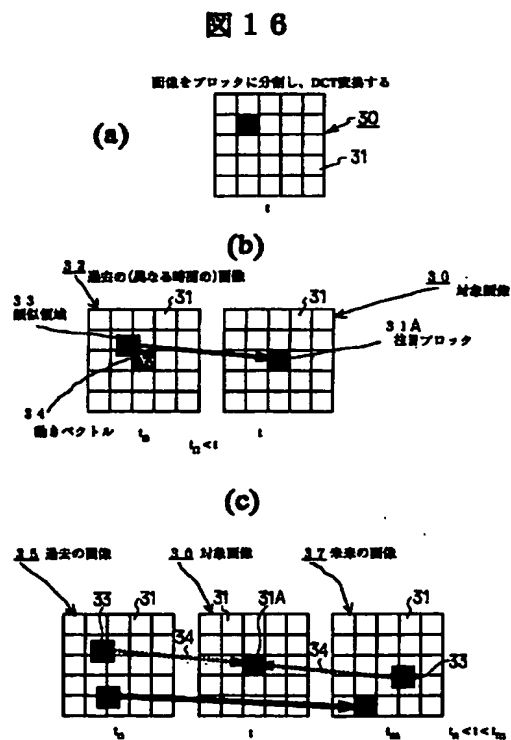
图 12



【图 14】



【图 16】



フロントページの続き

(72)発明者 浜田 洋
東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(72)発明者 外村 佳伸
東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内